

Design and Implementation of an Agent-Based Intermediary Infrastructure for Electronic Markets

Gaurav Tewari, Pattie Maes
MIT Media Lab
E15-305, 20 Ames Street
Cambridge, MA 02142 USA
Tel: +1-617-253-7441
{gtewari, pattie}@media.mit.edu

ABSTRACT

This paper describes MARI (Multi-Attribute Resource Intermediary), a project which proposes to improve online marketplaces, specifically those that involve the buying and selling of non-tangible goods and services. MARI is an intermediary architecture intended as a generalized platform for the specification and brokering of heterogeneous goods and services. MARI makes it possible for both buyers and sellers alike to more holistically and comprehensively specify relative preferences for the transaction partner, as well as for the attributes of the product in question, making price just one of a multitude of possible factors influencing the decision to trade. Ultimately, we expect that the ability to make such specifications will result in a more efficient, richer, and integrative transaction experience.

Keywords: Electronic Commerce; Software Agents; Intermediaries; Highly Mediated Communications; Product Brokering; Merchant Brokering; Negotiation; Utility Theory; Electronic Markets.

1 INTRODUCTION

1.1 Overview

This paper describes MARI (Multi-Attribute Resource Intermediary) – a research project being conducted in the Software Agents Group at the MIT Media Lab [2]. MARI proposes to radically improve online marketplaces, specifically those that involve the buying and selling of non-tangible goods and services. MARI is an agent-based intermediary architecture intended as a generalized platform for the specification and brokering of heterogeneous goods and services. MARI lies at the intersection of three general areas of research -- multi-agent systems, highly-mediated communications, and electronic commerce. Of these three general areas, MARI is positioned primarily from a “highly-mediated communications” perspective.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or to publish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

EC'00, October 17-20, 2000, Minneapolis, Minnesota.
Copyright 2000 ACM 1-58113-272-7/00/0010...\$5.00.

1.2 Research Focus and Objectives

State of the art online marketplaces, such as Chemdex [29], Priceline [30], Elance [31], etc. accentuate the importance of price in determining which seller the buyer transacts with and vice versa. This results in a static, impersonal bidding experience, and in an inability for the buyer and seller to transcend price as the only negotiative dimension. There is simply no means to convey the full “value proposition” of the holistic product offering. Online auction systems have a tendency to foster a spirit of adversarial competitiveness in the buying process. In systems such as Ebay [16] and Amazon Auctions [17], not only must a buyer first undergo the burden of uniquely identifying the exact product she is seeking, but furthermore, she must then enter into an inflexible and antagonistic bidding interplay with the seller. MARI attempts to overcome these limitations. MARI makes it possible for both buyers and sellers to more holistically and comprehensively specify relative preferences for the transaction partner, as well as for the attributes of the product in question, making price just one of a multitude of possible factors influencing the choice of trading partner and the decision to trade. MARI is unique in the sense that it allows both the buyer as well as the seller to exercise control. By allowing each party to choose and implicitly associate weights with relevant features from the underlying ontology, MARI makes it possible to take into account subtle differences in characteristics of each party, so as to facilitate a more accurate match.

2 RELATED WORK

Unlike most online shopping systems which generally operate in only one stage of the online shopping process [4], MARI operates in three core stages -- namely product brokering, merchant brokering, and negotiation -- to provide a unified experience that better facilitates economically efficient and socially desirable transactions. MARI amalgamates features of the ‘Market Maker’ [11] and ‘Tête-à-Tête’ [12] projects at the Media Lab, and extends these to create a more comprehensive solution. In particular, MARI builds upon multi-attribute utility theory formulations, as introduced in Tête-à-Tête, to model relative user preferences and quantify tradeoffs.

MARI relates to first generation price-comparison systems such as BargainFinder [7] and Jango [8], but goes much further than the rudimentary functionality afforded by such tools. MARI goes beyond just bid and ask prices to include the attributes of the transaction parties as dimensions for consideration and differentiation.

MARI relates to second generation value comparison shopping systems such as Personalogic [5], MySimon [13], and the Frictionless ValueShopper [6] in that it offers an advanced decision support engine, based upon multi-attribute utility theory, that meaningfully facilitates the exchange of complex and heterogeneous products. It differs from these systems in that it (i). allows both parties (buyers and sellers) to search for an optimal transaction partner, and (ii). it automates the match making between buyers and sellers. Further, MARI supports a non-linear and iterative user-interaction model, that accurately reflects the true nature of real-life transactions.

MARI relates to online negotiation systems and auctions, such as Kasbah [9] and AuctionBot [10], and commercial systems provided by Moai [32], TradingDynamics [33] and others. It differs from them in proposing an integrative negotiation protocol and interaction model. This model, based upon bilateral argumentation, embodies an appropriate blend of formality and efficiency, and provides an alternative to the adversarial competitiveness of online auctions.

Additionally, MARI relates to work in operations research done in the domain of dynamic pricing of inventories [19, 20, 21]. Specifically, we address the issue of how sellers should dynamically shift their valuations when demand is price sensitive and stochastic, and the seller's objective is to maximize expected revenues. Moreover, our algorithms for matching buyers and sellers are fundamentally based on flow algorithms as encountered in combinatorial optimization and network theory [22].

Finally, MARI builds on work done in the area of market-oriented allocation mechanisms [23, 24, 26]. We build upon economic theory in general, and game theory in particular, to formulate our problem in economics terminology [25, 15] with optimization heuristics, such as maximization of aggregate surplus, that derive directly from the literature.

3 RESEARCH DOMAIN

3.1 Functional Overview

MARI embodies a trend, expected to be key to the electronic marketplaces of tomorrow. Specifically, we believe that negotiations will be highly complex and participants will engage in integrative negotiation over various aspects of a transaction, price being only one of many considerations.

MARI represents a general purpose architecture that is capable of supporting multiple sellers and buyers within multiple product domains. For the purposes of our project we envision deploying the MARI infrastructure in the context of a "services marketplace" in which language translation services are bought and sold. Hence, MARI is specifically encoded with a "language translation service" ontology and suitable complementary data.

3.2 Services Marketplaces

A substantial motivation for the choice of "services marketplaces" in general, and a language-translation marketplace in particular, as the application domain for this project lies in our belief [27] that in the future, as people become increasingly networked, it will be easier for individuals to mutually help one another. People who

share each other's notions of quality and other such intangible attributes are in a much better position to be helpful to one another, at least until agents become truly "intelligent," and can be the ones helping people with complex tasks. For instance, given the extremely rudimentary capabilities of current state of the art automated translation systems [28], a (networked) person would be much better off if she could receive help with a language translation problem from a *human* expert located somewhere else. In this context, it is reasonable to postulate that the importance of technologies that mediate communications amongst end users will emerge as being of critical importance.

3.3 The Language Translation Marketplace

In order to be able to participate in the MARI marketplace, a "seller" creates a "selling agent" that is aware of its owner's level of expertise, availability, compensation expectations, and other special constraints, such as requirements for the buyer. Similarly, a "buyer" creates a "buying agent" that understands the exact needs of its owner such as degree of expertise desired, time-sensitivity or urgency with which information is needed, range and type of compensation that the buyer is willing to offer the seller, and other special constraints, such as minimum requirements on the seller's reputation level. Additionally, the buying (selling) agents also encapsulate information on how different qualified sellers (buyers) can be rank ordered in degree of relative preference. Subsequently, the "market" automatically matches buyers and sellers. Once a match has been made, other media (such as email, cell-phone, as well as richer media) may be employed to implement the "knowledge transfer relationship" in practice.

A benefit of focusing specifically on "services marketplaces" and "information goods" is that it allows us to concentrate more on the attributes relevant to the parties attempting to engage in the transaction without getting overwhelmed by the material details inherent to the product itself. For many information goods, the characteristics of the seller actually serve to define the good itself. For instance, in a language translation marketplace, the fact that the seller of the translation service is considered an expert, and has a high reputation rating associated with her to substantiate this claim, implicitly conveys the nature and quality of the "good," in this case the translation service. Indeed, one can argue that in the context of services marketplaces, in which the service being bought and sold lacks tangible manifestation and is hence not as easily susceptible to objective evaluation, the ability to be able to ontologically segregate and prioritize the various subtle impinging factors gains significance and relevance. Hence, the choice of services marketplaces, in which intangible services and information are bought and sold, is an appropriate and fitting choice as the target application domain for this project.

The electronic services marketplace embodied within MARI allows us to instantaneously match buyers and sellers in real-time. Ultimately, we hope, market forces will push the system towards an equilibrium where the time and efforts of a true expert are optimally used for just those questions that cannot easily be answered by anyone else. Resources will be bid up or down to reflect their true worth based upon a continuously updating balance between supply and demand of the scarce resource, and will thus be optimally allocated.

4 IMPLEMENTATION

4.1 Algorithms and Technologies Used

4.1.1 Overview All interaction between a given user¹ (buyer or seller) and MARI is mediated by the User Interface Manager (UIM) (see Figure 4). When a user initially logs onto MARI, he or she must specify whether her intent is to put a product or service up for sale or to purchase a product or service. Depending upon the user's choice, the UIM presents the appropriate interface to the user, such that the system is able to adequately capture all relevant parameters. The buyer's and seller's interfaces are, in fact, remarkably similar since, in both cases, our intent is to gauge the user's multi-attribute utility function so as to be able to accurately assess how the user would value products he or she has not explicitly seen or "rated" before. Being able to make such inferences is integral to the successful functioning of MARI's core matching algorithm.

MARI's market structure most closely resembles a "monopolistic competition" -- each seller has the ability to differentiate her products or services from those of other sellers². The market structure is monopolistic in the sense that each seller has the ability to set her own price, rather than merely accept the prevalent "market price" as under perfect competition and, thus, can be said to exercise market power. On the other hand, each seller must still compete, in terms of price and the range of products offered, with other sellers since they are all effectively trying to find transaction partners from a common underlying set of buyers. Moreover, there are no barriers to entry, and new sellers are free to enter the market. In this way, the market structure also resembles that of a competitive industry.

4.1.2 Capturing User Preferences Each distinct buyer or seller is represented within MARI by an agent. The "buyer agent" embodies the buyer's revealed preferences with respect to the desired resource. Similarly, "seller agents" embody the preferences and interests of sellers. Each agent is customized to the needs and desires of its owner, and attempts to advocate on the owner's behalf when finding suitable transaction partners. These agents are then used by the system to coordinate the preferences and interests of each party involved. MARI's interaction with the user, via which the user agent is initialized, can be decomposed into several steps, enumerated below. Each ontology-specific attribute has a predefined "default" value associated with it, and the user can accept or override these defaults.

Step 1). Specifying the Ideal Offer (see Figure 1): The user specifies a "referential" or "preferred" configuration, or *offer*, which consists of specific product and transaction partner attribute values, as derived from the underlying domain ontology. The user can modify which attributes are *fixed* and which are *flexible* and must also associate a monetary valuation ("bid" or "ask") with this offer (referred to as the *pbsvalue*).

The attributes of any given product can be classified as being either fixed or flexible. A fixed attribute is one whose value, as specified

¹ We use the term "user" to refer specifically to a buyer or seller. By contrast, we use the term "market maker" to refer to the system administrator who instantiates MARI within the context of a specific product domain.

² Since we specifically focus on complex products and services that consists of multiple, often non-tangible, attributes (such as seller reputation, for example), one could argue that the merchant offerings are differentiated a priori. Indeed, one can reasonably argue that in such product domains, it is extremely difficult, if not simply infeasible, for a given seller to perfectly replicate another seller's product offering.

by the user, is used for transaction party *qualification*. By contrast, flexible attributes have associated ranges, and are used for transaction party valuation. For instance, in the example of language translation services (buyer's perspective), the number of words to be translated could be a fixed attribute, while the reputation of the seller, the degree of expertise of the seller, and the amount of time within which the translation will be completed could be flexible attributes. Each fixed attribute has a predefined set of permissible values, and the user must select acceptable values from this set. For instance, the permissible values for 'number of words to be translated' might be the set of non-negative integers.

Welcome Pattie:

You have Chosen to Buy Spanish to English Language Translation Service. Please Specify Values for the following attributes (Click on attribute name to see explanation) :

Number of Words	
Possible Range	More than 0 words
Your Preferred Value	400
<input checked="" type="radio"/> Your preferred value is flexible	<input type="radio"/> Your preferred value is rigid

Seller Reputation	
Possible Range	From 1 [Worst] to 10 [Best]
Your Preferred Value	6
<input checked="" type="radio"/> Your preferred value is flexible	<input type="radio"/> Your preferred value is rigid

Seller Expertise	
Possible Range	From 1 [Worst] to 5 [Best]
Your Preferred Value	3
<input checked="" type="radio"/> Your preferred value is flexible	<input type="radio"/> Your preferred value is rigid

Task Completion Time	
Possible Range	From 10 minutes to 100 minutes
Your Preferred Value	30 minutes
<input checked="" type="radio"/> Your preferred value is flexible	<input type="radio"/> Your preferred value is rigid

Your Bid for your Preferred Values	\$39	99
------------------------------------	------	----

Figure 1: Specifying a "Referential" Configuration or "Offer"

Step 2). Gathering Ranges of Flexible Attributes (see Figure 2): Having specified which attributes ought to be considered flexible and which ought to be fixed, a user must also associate a permissible range of values with each flexible attribute. Further, in order to exercise some constraint on automatically generated bids and asks, the user must also specify the range (defined by a pair of highest and lowest endpoints, referred to as *maxvalue* and *minvalue*, respectively) of permissible valuations. This range corresponds to valuating the best and least qualified transaction partner, respectively.

Step 3). Inferring Attribute Weights: Once the User Interface Manager (UIM) has captured the above parameters, it automatically infers relative weights to be associated with each flexible attribute. The existence of weights is indicative of the fact that the user

associates different degrees of importance with the various attributes, when evaluating a given offer.

Asking a user to explicitly specify weights for each attribute would no doubt be the most accurate and transparent technique. However, doing so imposes additional burden and tedium on the user. Moreover, it is not at all clear whether users themselves are able to accurately quantify these numerical relative preferences. MARI automatically derives relative “weights” for flexible attributes by using the heuristic that an attribute’s weight or relative importance is proportional to how constrained the range of permissible values is, relative to the ranges of other flexible attributes [18]. A tightly constrained range indicates that the user is relatively unwilling to compromise and hence the attribute is relatively more significant to her. With this in mind, we use the following formula to calculate the numerical weight factor to be associated with a given (flexible) attribute, p :

$$Weight(p) = \frac{1 - \left(\frac{\text{Permissible Range of } p}{\text{Possible Range of } p} \right)}{\sum_i \left[1 - \left(\frac{\text{Permissible Range of } i}{\text{Possible Range of } i} \right) \right]} \quad (i \text{ ranges over all flexible attributes})$$

Figure 2: Specifying Ranges for Flexible Attributes or

4.1.3 Modeling User Utility Functions Since MARI is fundamentally an infrastructural tool, we expect that each “market maker” will create an instantiation of MARI specific to their product domain. When instantiating MARI, the market maker must specify the product ontology as well as a set of parameters which determine how user utility functions are modeled and what heuristics are optimized in the match making process.

The market maker initializes the Buyer or Seller Valuation Manager (B/SVM) (see Figure 4), whose purpose is to gather sufficient information from the user so as to be able to accurately infer how her (uni-dimensional) utility might change as each flexible attribute varies over its permissible range. Doing so enables us to accurately assess how the user would value product offerings and transaction partners that have not been explicitly seen or “rated” before. The process of automatically valuating a potential transaction partner then simply becomes a matter of taking a weighted sum of uni-dimensional utility functions.

Based upon the market maker’s configuration parameters, the Valuation Managers model the user’s utility function as follows:

Step 1. Visually Selecting Utility Functions: When first instantiating MARI, the market maker is required to initialize the Valuation Function Generalizer (VFG). The purpose of the VFG is to model the user’s utility for various ontology attributes by allowing the market maker to visually associate a generic (pre-defined) mathematical function with each flexible attribute [12] (see Table 1 and Figure 3). Of course, users have the option of being able to override these “default” values during the offer specification process.

Figure 3: Visually Associating Utility Functions with Flexible Attributes

Step 2. Quantifying Utility Functions: Using the generalized equation form of the utility function selected by the market maker (see Table 1), in conjunction with the $pbsvalue$, $maxvalue$, and $minvalue$ parameters specified by a given user (see Figures 1 and 2), the Valuation Function Generalizer (VFG) is able to compute a mathematical approximation to the utility function corresponding to each flexible attribute. The polynomial used to represent the function can be at most two (quadratic). Higher order polynomial approximations can subsequently be created by the Valuation

Function Trainer (VFT) using Lagrange interpolation, as described in Step 3 below.

Table 1: MARI’s Predefined Generic Utility Functions (x is a place-holder variable for a value of the flexible attribute over its permissible range)

Name	Functional Generalization (a, b, c represent arbitrary, non-negative constants)	Graph Shape
UF ₁	(ax ± b)	
UF ₂	(ax ² ± bx ± c)	
UF ₃	(-ax ² ± bx ± c)	
UF ₄	(-ax ± b)	
UF ₅	(ax ² ± bx ± c)	
UF ₆	(-ax ² ± bx ± c)	
UF ₇	x < (midpoint of permissible rang ax ± b) : (-ax ± b)	
UF ₈	x < (midpoint of permissible rang ax ² ± bx ± c) : (ax ² ± bx ± c)	
UF ₉	x < (midpoint of permissible rang ax ² ± bx ± c) : (-ax ² ± bx ± c)	
UF ₁₀	x < (midpoint of permissible rang ax ± b) : (ax ± b)	
UF ₁₁	x < (midpoint of permissible rang ax ² ± bx ± c) : (ax ² ± bx ± c)	
UF ₁₂	x < (midpoint of permissible rang ax ² ± bx ± c) : (-ax ² ± bx ± c)	
UF ₁₃	x = (high end point of permissible 1 : 0	
UF ₁₄	x = (high end point of permissible range) ? 0 : 1	
UF ₁₅	x = 1	
UF ₁₆	x = (low end point of permissible : 0	
UF ₁₇	x = (low end point of permissible : 1	

For example, let us assume that a given buyer is willing to accept a “seller reputation” ranging from 6 to 10. Assume that in her “referential offer” the buyer specifies a preferred value of 6. Further, say the market maker has pre-associated UF₂ with this flexible attribute as it varies over its range – the choice of this utility function would reflect the fact that the buyer is willing to bid higher as the seller’s reputation increases, and that her valuation increases exponentially as reputation approaches the maximum possible. In

this case we can derive the equation³ which captures the change in the buyer’s utility as reputation varies, as:

$$UF_2(x) = \left(\frac{\text{maxvalue} - \text{pbsvalue}}{(x_{hi} - x_{low})^2} \right) x^2 + \left(\frac{(-2)(\text{maxvalue} - \text{pbsvalue})(x_{low})}{(x_{hi} - x_{low})^2} \right) x + \left(\text{pbsvalue} + \frac{(\text{maxvalue} - \text{pbsvalue})(x_{low})^2}{(x_{hi} - x_{low})^2} \right)$$

Where:

x_{low} = the value of the attribute specified in the referential offer (i.e. 6);

x_{hi} = high endpoint of the permissible range (i.e. 10). TechNotes: 2 pages maximum

Step 3). Refining Utility Functions through Revealed Preferences: In this stage, the Valuation Manager attempts to fine-tune the rough utility function captured by the VFG. During system initialization, the VFG invokes the Valuation Function Trainer (VFT), which requires that the market maker iteratively “train” the system. At that time, the market maker is asked to explicitly “value” hypothetical product offerings strategically chosen to representatively span the space of all relevant product offerings. Essentially, the preferences expressed by the market maker are taken to be a benchmark set of “reasonable” preferences. Using these revealed preferences in conjunction with offer data that is specific to a particular user, the VFT facilitates the construction of a (iteratively refined) piecewise, linear approximation of the user’s utility function.

If there are n flexible attributes, then the user’s utility function can be visualized as an n dimensional hyper plane in $(n+1)$ dimensions (where the $(n+1)^{st}$ dimension is the numerical “monetary” valuation associated with each point on the hyper plane). The VFT uses Lagrange Interpolation, in conjunction with an iterative scheme we shall refer to as “delta scaling,” in order to determine higher degree polynomial approximations for utility functions.

In general, we can use Lagrange Interpolation to approximate the value of any $(n-1)^{st}$ degree polynomial, f , at any arbitrary point, x , provided that we already know the values of the function (f_0, f_1, \dots, f_{n-1}) at n distinct points (x_0, x_1, \dots, x_{n-1}), by using the following expression:

$$f(x) = f_0 \frac{(x-x_1)(x-x_2)\dots(x-x_{n-1})}{(x_0-x_1)(x_0-x_2)\dots(x_0-x_{n-1})} + \dots + f_{n-1} \frac{(x-x_0)(x-x_1)\dots(x-x_{n-2})}{(x_{n-1}-x_0)(x_{n-1}-x_1)\dots(x_{n-1}-x_{n-2})}$$

The term “delta scaling” refers to the technique by which the VFT picks points in the attribute space to be valued by the market maker. The algorithm goes through iterations referred to as “delta phases.” In each delta phase, the algorithm unidimensionally varies the value of a single flexible attribute (by “delta”) and asks the market maker to explicitly associate a valuation with the feature set, thus effectively obtaining an additional data point relevant to this

³ This function is derived using standard algebraic techniques along with special properties of quadratic functions. In particular, we have used the fact that the global minima of a quadratic, of the form

$$y = ax^2 + bx + c, \text{ occurs at } \frac{-b}{2a}, \text{ that the quadratic function}$$

corresponding to UF₂ is monotonically increasing, and that the user has already revealed two data points, (x, y) , on the curve: $(x_{low}, \text{pbsvalue})$ and $(x_{hi}, \text{maxvalue})$.

attribute. Subsequently, the algorithm performs Lagrange interpolation to compute a higher degree (uni-dimensional) utility function for the attribute. In a given delta phase, the algorithm does this for each attribute, thereby improving the degree of each attribute's associated utility function by at least one. The market maker thus provides MARI with an internal model of reasonable user preferences. This model can be thought of as a "template," that is subsequently adapted to user-specific data.

4.1.4 Delineating Transaction Partners MARI operates by using the notion of "market cycles." At the beginning of every market cycle, MARI goes through two phases. In the first phase, for each buyer, MARI identifies the sellers who are qualified to meet the buyer's request. This corresponds to the subset of sellers who are able to satisfy the buyer's fixed attribute requirements. In the second phase, MARI uses its internal mathematical approximation of the buyer's and sellers' utility functions to calculate "bids" and "asks." For instance, given an arbitrary seller, s , we can compute buyer b 's valuation or "bid" for s as:

$$\text{Valuation}_{b,s} = \sum_i f_i(x_{i,s}) * w_i$$

Where:

- i , ranges over all flexible attributes;
- f_i is the buyer's revealed utility function corresponding to attribute i ;
- $x_{i,s}$ is the seller-specific value of attribute i ;
- w_i is the weight associated with attribute i by the buyer.

Then, for each buyer, MARI evaluates the "cost" that would be incurred if the buyer were to engage in a transaction with any of the qualified sellers. Currently, we take this "cost" to be equal to the "bid-ask spread," which can be interpreted as the aggregate surplus⁴ [14, 15] that the two parties would derive if the transaction were to take place. We use this metric of "cost" since our indicator of the "goodness" of an allocation is welfare, which, in this case, is measured by the surplus that the allocation generates.

Subsequently, we can conveniently formulate the problem of optimally pairing up buyers and sellers as a "matching" problem. Mathematically, we can represent the state of the marketplace as a graph, G , in which sellers and buyers represent nodes. We refer to the set of buyers and seller as B and S , respectively, and to the set of arcs as A . Each individual buyer node $b \in B$ is connected to a subset of seller nodes $S' \subseteq S$, via some arc (b, s) with associated arc "cost" c_{bs} . Given this formulation, our goal is to find a sub-graph $G' \subseteq G$, such that the sub-graph represents a feasible⁵ pairing of buyers and sellers with the largest overall "cost" (surplus), defined as the sum of the costs of its constituent arcs. To accomplish this, our solution strategy mirrors that of a (modified)⁶ *minimum* cost

⁴ Aggregate surplus is the sum of consumer and producer surplus. Consumer surplus is defined as the difference between the amount a consumer is willing to pay for a good and the amount she actually pays. Producer surplus is defined as the difference in the market price the producer receives for a good and the marginal cost incurred in its production [15].

⁵ In this case the "feasibility" condition maintains that for a given buyer, the seller should be *qualified* to serve the buyer *and* that the buyer should meet the *qualification* criteria, if any, specified by the seller. Moreover, the buyer's bid can be no less than the seller's ask.

⁶ Since we are actually trying to *maximize* the sum of our costs (aggregate surplus), we redefine costs in the min cost flow formulation to be the

flow problem. With this formulation the matching problem can now be expressed as the following linear program [22]:

$$\begin{aligned} & \text{Minimize } \sum_{(i,j) \in A} c_{ij} x_{ij} \\ & \text{subject to the constraints:} \\ & \sum_{\{j:(i,j) \in A\}} x_{ij} = 1 \quad \text{for all } i \in S, \\ & \sum_{\{j:(i,j) \in A\}} x_{ji} = 1 \quad \text{for all } i \in B, \\ & x_{ij} \geq 0 \quad \text{for all } (i,j) \in A. \end{aligned}$$

We solve this to identify buyer-seller pairings for which the aggregate surplus of transaction parties is globally maximized. The "clearing price" for any given transaction pair is, by default, set at the midpoint between the original bid and ask prices, thereby equally dividing the surplus between the buyer and the seller. The market maker can, however, modify this distribution of surplus, choosing to retain the bid-ask spread as operating profit for instance.

4.2 System Architecture

Overall, the MARI system architecture consists of the following major functional components (see Figure 4):

1). *User Interface Manager (UIM)*: Controls the (HTML) interface that is presented to the user. The UIM allows the user to specify and initiate a buy or sell request, to examine the status of previous requests, and to view market statistics and history. The Valuation Managers, discussed below, are invoked by the UIM during the course of initializing a user request to buy or sell. The UIM ensures that all relevant parameters are collected from a user in the context of any request, and that only valid requests are propagated.

2). *Buyer Valuation Manager (BVM)*: Gathers sufficient information from a potential buyer so as to be able to accurately infer the buyer's valuation for previously unseen products.

2.1). *Valuation Function Generalizer (VFG)*: Models a buyer's utility for multiple attributes by allowing the market maker to select from generic, pre-defined mathematical functions.

2.2). *Valuation Function Trainer (VFT)*: Fine-tunes the rough utility function captured by the VFG by giving the market maker the option of iteratively "training" the system. Essentially, the VFT facilitates the construction of an (iteratively refined) piecewise, linear approximation of the buyer's utility function.

3). *Seller Valuation Manager (SVM)*: Gathers information from a potential seller so as to be able to accurately infer the seller's valuation for previously unseen products. Like the BVM, the SVM works by having the market maker initialize the VFG and VFT.

4). *Market Cycle Manager (MCM)*: Manages and enforces market cycles. The frequency of market cycles is a system variable that must be pre-specified, or can be set to be triggered by the simultaneous presence of certain (pre-specified) environmental conditions (such as number of users currently waiting to be matched).

negative of the computed surpluses. Minimizing the sum of the *negatives* of the original quantities is equivalent to maximizing the sum of the original quantities.

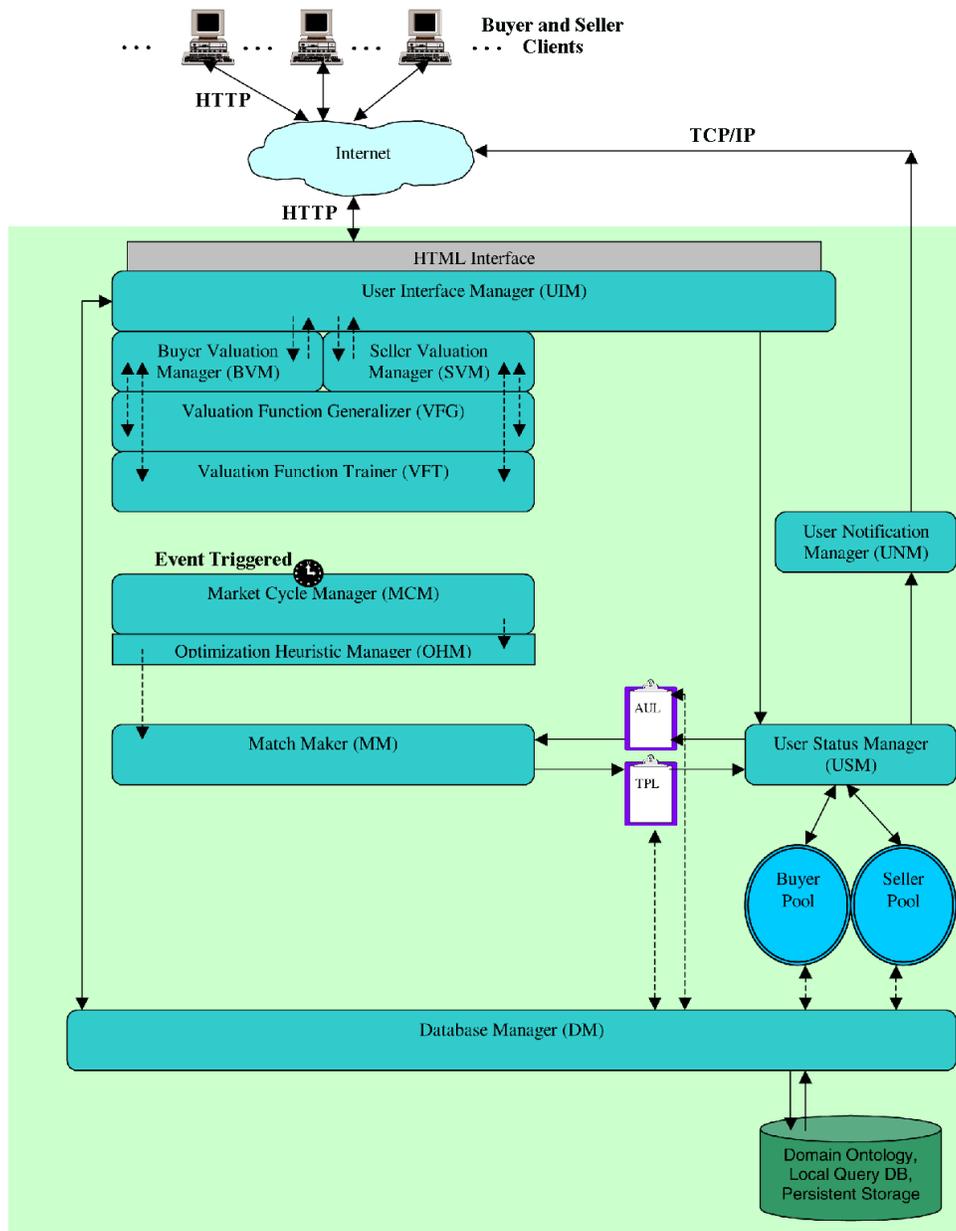


Figure 4: System Schematic

4.1). *Optimization Heuristic Manager (OHM)*: Allows the specification of which optimization heuristic ought to be employed. For instance, in some cases, the market maker's aim might be to maximize the *number* of buyers and sellers matched, while in other cases one may want to maximize the minimum surplus amongst all transaction partners (Rawlsian approach [15]), etc. The OHM enforces the specified optimization heuristic throughout the system by setting key *global* parameters appropriately, in a mutually consistent fashion. These parameters can then be referenced by other modules in the process of optimally pairing transaction partners.

The MCM invokes the Match Maker (MM) at the start of every market cycle. The MCM also ensures that buying and selling requests are time-stamped and queued appropriately so that precedence and priority relationships can be established if needed. At the end of every market cycle, the MCM examines the results output by the MM and notifies the User Status Manager (USM) and User Notification Manager (UNM), so as to update the status of user requests, as appropriate.

5). *Match Maker (MM)*: Invoked by the MCM, the MM optimally by the MM is specified by the MCM at the time of invocation. The

MM gets the IDs of “active” users to be included in the matchmaking process from the User Status Manager (USM).

6). *User Status Manager (USM)*: The USM monitors the status of each user. The USM keeps track of which requests are “active” and ought to be included in the matchmaking process in any given market cycle. For instance, the market maker may require that any given user should remain active for at least n market cycles, or, perhaps, that a given user should remain active until she is involved in market cycles with an *aggregate* of at least m other users, at least m_1 of whom are sellers and m_2 are buyers.

7). *User Notification Manager (UNM)*: The UNM notifies a given user of the outcome of their buying or selling request once a definitive outcome has been established or the time permitted by the user has expired. “Notification” can be “active” (sending an e-mail to the user) or “passive” (writing the outcome to a local database which is queried when the user logs in to check the status of her request).pairs up buyers and sellers. The exact optimization heuristic to be used.

8). *Database Manager (DM)*: The DM presents each of the above components with an interface to a back-end database, thus abstracting away the specific details by which data is stored and retrieved from the rest of the system.

9). *Active User List (AUL)*: At the beginning of each market cycle, the Match Maker (MM) *reads* the AUL to identify “active” buyers and sellers who need to be matched. The User Status Manager (USM) *updates* the AUL when a new user enters the system and at the end of each market cycle.

10). *Transaction Partner List (TPL)*: The TPL is a list of transaction partners as determined by the Match Maker (MM) at the termination of the most recent market cycle.

The AUL, TPL, and Buyer and Seller Pools effectively comprise a system “log” that capture the state of the system at the end of a Market Cycle. In the event of a system failure, we revert back to the last logged state.

As of now, we have completed implementing a prototype of the system using an HTML front end, driven by Java servlets to manage content and user interaction, and integrated with a back-end SQL database for persistent storage. Additionally, we are in the process of writing DTDs for product ontologies and buyer and seller profiles, since these will be created, maintained and stored in XML.

5 FUTURE WORK

As of now, we have precisely defined MARI’s design framework and functional modules, and have delineated the core algorithms that will be used in gathering user utility functions, “valuating” potential transaction partners, and optimally matching buyers and sellers. In the near future, we expect to actually deploy the MARI infrastructure to build a language translation marketplace within the context of the visionary Nation1 virtual youth community established by the Media Lab [3]. We expect that actually deploying our system in such a setting and using it to broker translation services will allow us to benefit from direct user feedback to address considerations such as privacy preservation, individual rationality, incentive compatibility, market liquidity, and stability of matchings (sensitivity analysis), as well as more mundane concerns such as speed, accuracy, data integrity, ease of use, and scalability.

As we further refine the MARI architecture and implementation, we expect to face a number of key questions. In particular, even though we have identified one set of models by which agents will interact and transaction partners will be determined, several issues remain to be addressed. We are keen to explore what algorithms and technologies our system can leverage in the process of information integration and representation, decision analysis, modeling and reasoning about utilities, heuristical learning and inference from user-interaction, facilitating inter-agent communication and negotiation, and attuning pre-existing knowledge bases in developing and managing shared product ontologies. Further, in the near future, we would like to explore the usage of standardized agent communication languages, we would like to see how machine learning techniques can be used in assisting decision support and negotiation, and will also facilitate dynamic alteration of bids and asks based upon stochastic demand and supply patterns over finite horizons [19, 20, 21].

Having completed a satisfactory implementation of the MARI infrastructure, we would next like to undertake simulations that employ different optimization heuristics, welfare metrics, and matching algorithms. We are curious to study how the quality of the outcome changes as we vary these parameters. Currently we are in the process of evaluating what additional concerns our system needs to address, and what kinds of simulations might be particularly compelling to undertake.

6 BIBLIOGRAPHY

- [1] MARI: <http://www.media.mit.edu/gtewari/MARI/>
- [2] Software Agents Group:
<http://agents.www.media.mit.edu/groups/agents/>
- [3] Nation1: <http://www.nation1.net>
- [4] Maes, P., Robert H. Guttman and Alexandros G. Moukas, “Agents That Buy and Sell.” *Communications of the ACM*, March 1999, Vol. 42, No. 3.
- [5] Personallogic: <http://www.personallogic.com>
- [6] Frictionless ValueShopper: <http://compare.frictionless.com>
- [7] Bargainfinder: <http://bf.cstar.ac.com/bf>
- [8] Jango: <http://www.jango.com>
- [9] Kasbah: <http://ecommerce.media.mit.edu/Kasbah/>
- [10] Wurman, Peter R., Michael P. Wellman and William E. Walsh. “The Michigan Internet AuctionBot: A Configurable Auction Server for Human and Software Agents.” In *Proceedings of the Second International Conference on Autonomous Agents (Agents-98)*, Minneapolis, MN, USA. May 1998.
- [11] Wang, David Yi. “Market Maker: An Agent-Mediated Marketplace Infrastructure.” Master of Engineering Thesis, Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA. June 1999.
- [12] Guttman, Robert H. “Merchant Differentiation through Integrative Negotiation in Agent-mediated Electronic Commerce.” Masters Thesis, MIT Media Laboratory, Cambridge, MA. September 1998.
- [13] MySimon Buyers Guide:
http://www.mysimon.com/consumer_resources/Buyers_Guides/index.anml

- [14] Pindyck, Robert S. and Daniel R. Rubinfeld. Microeconomics. Prentice-Hall, New Jersey.1997.
- [15] Varian, Hal R. Intermediate Microeconomics. W. W. Norton & Company, New York. 1999.
- [16] Ebay: <http://www.ebay.com>.
- [17] Amazon Auctions: <http://www.amazon.com/auctions>.
- [18] Morris, J. and P. Maes. "Negotiating Beyond the Bid Price." To appear, Workshop Proceedings of the Conference on Human Factors in Computing Systems (CHI 2000), The Hague, The Netherlands. April 1-6, 2000.
- [19] Gallego, Guillermo and Garrett Ryzin. "Optimal Dynamic Pricing of Inventories with Stochastic Demand over Finite Horizons." Management Science, 40(8), p.999-1020. August 1994.
- [20] Bitran, Gabriel R. and S. Mondshein. "Periodic Pricing of Seasonal Products in Retailing." Management Science, 45(8), p. 64-79. January 1997.
- [21] Raman, Kalyan and R. Chatterjee. "Optimal Monopolistic Pricing under Demand Uncertainty in Dynamic Markets." Management Science, 41(1), p. 144-162. January 1995.
- [22] Ahuja, Ravindra K., Thomas L. Magnanti and James B Orlin. Network Flows: Theory, Applications and Algorithms. Prentice-Hall, New Jersey.1993.
- [23] Boutilier, C., Y. Shoham and M. Wellman. "Economic Principles of Multi-Agent Systems" (editorial). Artificial Intelligence, 94, p.1-6. 1997.
- [24] Sandholm, T. and Fredrick Ygge. "On the Gains and Losses of Speculation in Equilibrium Markets." In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, p. 632-638, Nagoya, Japan. 1997.
- [25] Fudenberg, D., and J. Tirole. Game Theory. MIT Press, Cambridge, MA. 1995.
- [26] Walsh, W., M. Wellman, P. Wurman, J. MacKie-Mason. "Some Economics of Market-Based Distributed Scheduling." In Proceedings of the Eighteenth International Conference on Distributed Computing Systems, p. 612-621. 1998.
- [27] Pattie Maes' statement: <http://pattie.www.media.mit.edu/people/pattie/statement.html>
- [28] Altavista online translation system: <http://world.altavista.com/>
- [29] Chemdex: <http://www.chemdex.com>
- [30] Priceline: <http://www.priceline.com>
- [31] Elance: <http://www.elance.com>
- [32] Moai: <http://www.moai.com>
- [33] TradingDynamics: <http://www.tradingdynamics.com>